td8-web

March 21, 2022

1 TD8 - web - Emanuele Marnati

[157]:
```
import numpy as np
import sys
import pandas
```

1.1 1. Write the corresponding matrix S, as explained during the class.

[158]:
```
S = [[0,0,0,0,0,0,0,0],
    [0,0,1,0,0,0,0,0,],
    [0,1,0,0,0,0,0,0],
    [0.5,0.5,0,0,0,0,0,0],
    [0,0,0,0.5,0,0.5,0,0],
    [0,0.5,0,0,0.5,0,0,0],
    [0,0.5,0,0,0.5,0,0,0],
    [0,0.5,0,0,0.5,0,0,0]]
```

[159]:
```
S_corr = [[0.125,0.125,0.125,0.125,0.125,0.125,0.125,0.125],
    [0,0,1,0,0,0,0,0,],
    [0,1,0,0,0,0,0,0],
    [0.5,0.5,0,0,0,0,0,0],
    [0,0,0,0.5,0,0.5,0,0],
    [0,0.5,0,0,0.5,0,0,0],
    [0,0.5,0,0,0.5,0,0,0],
    [0,0.5,0,0,0.5,0,0,0]]
```

1.2 2. Compute, from matrix S, matrix G = delta*S + (1 − delta)E, for delta = 0.85 and a teleportation matrix E, whose rows consist of the vector u = (1/n, . . . , 1/n).

[160]:
```
E = [[0.125,0.125,0.125,0.125,0.125,0.125,0.125,0.125],
    [0.125,0.125,1,0.125,0.125,0.125,0.125,0.125,],
    [0.125,1,0.125,0.125,0.125,0.125,0.125,0.125],
    [0.125,0.125,0.125,0.125,0.125,0.125,0.125,0.125],
    [0.125,0.125,0.125,0.125,0.125,0.125,0.125,0.125],
    [0.125,0.125,0.125,0.125,0.125,0.125,0.125,0.125],
    [0.125,0.125,0.125,0.125,0.125,0.125,0.125,0.125],
```

```
    [0.125,0.125,0.125,0.125,0.125,0.125,0.125,0.125]]
```

[161]:
```
delta = 0.85
```

[162]:
```
(1 - delta)
```

[162]: 0.15000000000000002

1.2.1 calcolo matrice delta * S

[163]:
```
for i in range(0,len(S_corr)):
    for j in range(len(S_corr[i])):
        #print(S[i][j]*delta)
        S_corr[i][j] = round(S_corr[i][j]*delta, 4)
        deltaS = S_corr
print(deltaS)
```

```
[[0.1062, 0.1062, 0.1062, 0.1062, 0.1062, 0.1062, 0.1062, 0.1062], [0.0, 0.0,
0.85, 0.0, 0.0, 0.0, 0.0, 0.0], [0.0, 0.85, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
[0.425, 0.425, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0], [0.0, 0.0, 0.0, 0.425, 0.0, 0.425,
0.0, 0.0], [0.0, 0.425, 0.0, 0.0, 0.425, 0.0, 0.0, 0.0], [0.0, 0.425, 0.0, 0.0,
0.425, 0.0, 0.0, 0.0], [0.0, 0.425, 0.0, 0.0, 0.425, 0.0, 0.0, 0.0]]
```

1.2.2 calcolo matrice (1-delta)E

[164]:
```
for i in range(0,len(E)):
    for j in range(len(E[i])):
        #print(S[i][j]*delta)
        E[i][j] = round(E[i][j]*(1-delta), 4)
        E_corr = E
print(E_corr)
```

```
[[0.0188, 0.0188, 0.0188, 0.0188, 0.0188, 0.0188, 0.0188, 0.0188], [0.0188,
0.0188, 0.15, 0.0188, 0.0188, 0.0188, 0.0188, 0.0188], [0.0188, 0.15, 0.0188,
0.0188, 0.0188, 0.0188, 0.0188, 0.0188], [0.0188, 0.0188, 0.0188, 0.0188,
0.0188, 0.0188, 0.0188, 0.0188], [0.0188, 0.0188, 0.0188, 0.0188, 0.0188,
0.0188, 0.0188, 0.0188], [0.0188, 0.0188, 0.0188, 0.0188, 0.0188, 0.0188,
0.0188, 0.0188], [0.0188, 0.0188, 0.0188, 0.0188, 0.0188, 0.0188, 0.0188,
0.0188], [0.0188, 0.0188, 0.0188, 0.0188, 0.0188, 0.0188, 0.0188, 0.0188]]
```

1.2.3 calcolo matrice G = deltaS - E_corr

[165]:
```
for i in range(0,len( S)):
    for j in range(len( S[i])):
        #print(S[i][j]* )
        deltaS[i][j] = round(deltaS[i][j]-E_corr[i][j], 4)
        G = deltaS
```

```
print(G)
```

```
[[0.0874, 0.0874, 0.0874, 0.0874, 0.0874, 0.0874, 0.0874, 0.0874], [-0.0188,
-0.0188, 0.7, -0.0188, -0.0188, -0.0188, -0.0188, -0.0188], [-0.0188, 0.7,
-0.0188, -0.0188, -0.0188, -0.0188, -0.0188, -0.0188], [0.4062, 0.4062, -0.0188,
-0.0188, -0.0188, -0.0188, -0.0188, -0.0188], [-0.0188, -0.0188, -0.0188,
0.4062, -0.0188, 0.4062, -0.0188, -0.0188], [-0.0188, 0.4062, -0.0188, -0.0188,
0.4062, -0.0188, -0.0188, -0.0188], [-0.0188, 0.4062, -0.0188, -0.0188, 0.4062,
-0.0188, -0.0188, -0.0188], [-0.0188, 0.4062, -0.0188, -0.0188, 0.4062, -0.0188,
-0.0188, -0.0188]]
```

1.3 3. Compute vector pi-greco, solution of the equation pi-greco = pi-greco*G, using the power method. Carry out the calculations for at least two iterations of the method.

[166]:
```
A = G
```

[167]:
```
v = np.array([[1],[0], [0],[0],[0],[0],[0],[0],])
```

[168]:
```
for i in range(2):
    print('iteration no.', i+1)
    temp= np.matmul(A,v)
    print(temp)
    v = temp/np.max(temp)
    print(np.max(temp))
    print(v)
```

```
iteration no. 1
[[ 0.0874]
 [-0.0188]
 [-0.0188]
 [ 0.4062]
 [-0.0188]
 [-0.0188]
 [-0.0188]
 [-0.0188]]
0.4062
[[ 0.21516494]
 [-0.04628262]
 [-0.04628262]
 [ 1.        ]
 [-0.04628262]
 [-0.04628262]
 [-0.04628262]
 [-0.04628262]]
iteration no. 2
[[ 0.08193481]
 [-0.05089237]
```

```
 [-0.05089237]
 [ 0.05415057]
 [ 0.38770547]
 [-0.05696465]
 [-0.05696465]
 [-0.05696465]]
0.38770546528803546
[[ 0.21133262]
 [-0.13126554]
 [-0.13126554]
 [ 0.13966934]
 [ 1.        ]
 [-0.14692764]
 [-0.14692764]
 [-0.14692764]]
```

[169]:
```
pi-greco = [[ 0.08193481],
 [-0.05089237],
 [-0.05089237],
 [ 0.05415057],
 [ 0.38770547],
 [-0.05696465],
 [-0.05696465],
 [-0.05696465]]
```

[170]:
```
pi-greco
```

[170]:
```
[[0.08193481],
 [-0.05089237],
 [-0.05089237],
 [0.05415057],
 [0.38770547],
 [-0.05696465],
 [-0.05696465],
 [-0.05696465]]
```

Thanks for the attention